

**AMENDMENTS TO THE CLAIMS**

Please amend the claims as follows.

1. (Currently Amended) A method for managing ~~system~~ resources of a computer system, comprising:
  - creating a container, wherein creating the container comprises allocating a first portion of a first resource of the computer system to the container, wherein the computer system comprises a plurality of sets of processors;
  - associating the container with a resource pool, wherein the resource pool is associated with one of the plurality of sets of processors and is allocated a portion of the first resource, wherein the associated container resides in the resource pool;
  - determining whether the first portion of the first resource allocated to the container is valid, wherein the first portion of the first resource allocated to the container is valid when the first portion of the first resource allocated to the container does not exceed the portion of the first resource allocated to the resource pool; and
  - activating the container only if the first portion of the first resource is valid, wherein activating the container enables at least one system user to use the container.
2. (Original) The method of claim 1, further comprising:
  - executing a project within the container once the container is active.
3. (Original) The method of claim 2, further comprising:
  - collecting statistics corresponding to the executing of the project in the container.
4. (Original) The method of claim 2, further comprising:
  - triggering an alert if the project executing within the container attempts to use more than the first portion of the first resource.
5. (Original) The method of claim 2, wherein the project does not use more than the first portion of the first resource while executing in the container.

6. (Original) The method of claim 2, wherein the project is placed in the container by a user listed on an access control list associated with the container.
7. (Currently Amended) The method of claim 1, further comprising:
  - allocating a second portion of the first resource to the container, if the first portion of the first resource allocated to the container is not valid;
  - determining ~~whether that~~ the second portion of the first resource allocated to the container is valid when the second portion of the first resource allocated to the container does not exceed the portion of the first resource allocated to the resource pool; and
  - based on determining that the second portion of the first resource allocated to the container is valid, activating the container ~~if the second portion of the first resource allocated to the container is valid~~.
8. (Currently Amended) The method of claim 1, further comprising:
  - allocating a first portion of a second resource of the computer system to the container;
  - determining whether the first portion of the second resource allocated to the container is valid.
9. (Original) The method of claim 1, further comprising:
  - deactivating the container, wherein deactivating the container comprises releasing the first portion of the first resource from the container.
10. (Original) The method of claim 9, further comprising:
  - transferring the project executing the container to a default container if the container is deactivated; and
  - executing the project in the default container.
11. (Original) The method of claim 1, further comprising:
  - modifying the first portion of the first resource after the container is activated.

12. (Currently Amended) The method of claim 11, wherein modifying the first portion of the first resource comprises modifying a container definition of the container ~~is modified using schedule change job functionality.~~
13. (Original) The method of claim 1, wherein creating the container comprises:  
defining a container name;  
specifying a minimum CPU requirement for the container;  
specifying a maximum physical memory limit; and  
specifying a maximum outgoing network bandwidth.
14. (Original) The method of claim 12, wherein creating the container further comprises:  
specifying a project associated with the container, wherein the project corresponds to a plurality of processes.
15. (Currently Amended) The method of claim ~~[[13]]~~14, wherein each of the plurality of processes is identified by the same identifier.
16. (Currently Amended) The method of claim 1, wherein the first resource is at least one selected from ~~[[the]]~~ a group consisting of ~~a central processing unit (CPU),~~ physical memory~~[[,]]~~ and bandwidth.
17. (Currently Amended) A ~~resource-management~~ computer system, comprising:  
a plurality of sets of processors;  
a first resource and a second resource;  
a first resource pool, wherein the resource pool is allocated a portion of the first resource and a portion of the second resource, and wherein the resource pool is associated with one of the plurality of sets of processors;  
a first container residing in the first resource pool, wherein the first container comprises a requirements specification for the first resource for the first container and a requirements specification for the second resource for the first container; and  
a management interface configured to:

~~verify~~ validate that the requirements specification for the first resource [[with]] does not exceed the allocated portion of the first resource, and  
~~verify~~ validate that the requirements specification for the second resource [[with]] does not exceed the allocated portion of the second resource.

18. (Original) The system of claim 17, further comprising:

a database configured to track:

allocation of the first resource;

allocation of the second resource;

the requirements specification of the first resource for the first container; and

the requirements specification of the second resource for the first container.

19. (Original) The system of claim 17, further comprising;

a second container residing in the first resource pool, wherein the second container comprises a requirements specification the first resource for the second container and a requirements specification for the second resource for the second container;

20. (Original) The system of claim 19, wherein the usage of the first resource and the second resource by the first container and the second container is determined using fair share scheduling.

21. (Original) The system of claim 17, wherein the management interface is configured to modify the requirements specification for the first resource for the first container.

22. (Currently Amended) The system of claim [[21]] 17, wherein the requirements specification for the first resource for the first container and the requirements specification of the second resource for the first container are included in a container definition of the first container ~~is modified using schedule change job functionality.~~

23. (Original) The system claim 17, further comprising:

a project configured to execute in the first container, wherein the project corresponds to a network-wide administrative identifier used to identify related processes.

24. (Original) The system of claim 23, wherein the amount of the first resource used to execute the project in the first container does not exceed the portion of the first resource allocated to the first container.
25. (Original) The system of claim 23, wherein the amount of the first resource used to execute the project in the first container does not exceed the requirements specification of the first resource for the first container.
26. (Original) The system of claim 23, wherein the management interface is configured to track usage of the first resource and the second resource by the project.
27. (Original) The system of claim 23, wherein the project is placed in the first container by a user listed on an access control list associated with the first container.
28. (Original) The system of claim 17, further comprising:  
a first management utility configured to manage the first resource; and  
a second management utility configured to manage the second resource,  
wherein the management interface is further configured to interface with the first management utility and the second management utility to manage the portion of the first resource and the portion of the second resource allocated to the resource pool.
29. (Original) The system of claim 17, wherein the management interface is further configured to discover the first resource and the second resource.
30. (Original) The system of claim 17, wherein the first container comprises:  
a container name;  
a minimum CPU requirement for the container;  
a maximum physical memory limit;  
specifying a maximum outgoing network bandwidth.

31. (Currently Amended) The system of claim 17, wherein the first resource is at least one selected from [[the]] a group consisting of a ~~central processing unit (CPU)~~, physical memory[[,]] and bandwidth.
32. (Currently Amended) A network system having a plurality of nodes, comprising:
- a plurality of sets of processors;
  - a first resource and a second resource;
  - a [[first]] resource pool, wherein the resource pool is allocated a portion of the first resource and a portion of the second resource, and wherein the resource pool is associated with one of the plurality of sets of processors;
  - a [[first]] container residing in the [[first]] resource pool, wherein the [[first]] container comprises a requirements specification for the first resource for the [[first]] container and a requirements specification for the second resource for the [[first]] container; and
- a management interface configured to:
- ~~verify~~ validate that the requirements specification for the first resource [[with]] does not exceed the allocated portion of the first resource, and
  - ~~verify~~ validate that the requirements specification for the second resource [[with]] does not exceed the allocated portion of the second resource,
- wherein the first resource is located on any one of the plurality of nodes,  
wherein the second resource is located on any one of the plurality of nodes,  
wherein the [[first]] resource pool is located on any one of the plurality of nodes,  
wherein the container is located on any one of the plurality of nodes,  
wherein the management interface executes on any one of the plurality of nodes.